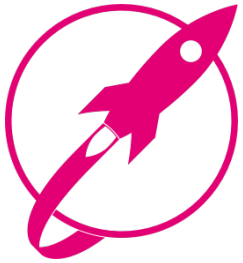


FH·W-S

iOS Programmierung

(mit Swift)

Peter Braun, Florian Bachmann & Andreas Wittmann
@pe_braun @florianbachmann @anwittmann

Deutsche Telekom AG
FHWS - Hochschule für angewandte Wissenschaften Würzburg-Schweinfurt

#FHWSSwift

Agenda

1. **Introduction** – Organisatorisches
2. **First iOS-Project** – Hello World, **First iOS-Project** – Still Hello World (now with Code 😄)
3. **Swift**, Wait!, What about Objective-C?, Why Swift?
4. **A (not so) Quick Tour**
5. **Documentation**
6. **The basics –** iOS Architecture & more
7. **User Interfaces –** View Controller, Auto Layout & Size Classes
8. **Storyboard & Segues**
9. **Tables & NavigationController**
10. **TabBarController**
11. **Notifications**
12. **PickerViews**
13. **Touches, Gestures, 3D Touch, Peek & Pop**
14. **ScrollView & StackViews**
15. **Networking** – JSON & Dependency Managers
16. **WebKit**
17. **Maps**
18. **Storage & Data persistency –** NSUserDefaults, NSKeyedArchiver & Core Data
19. **ObjC**

Today

Maps

CoreLocation & MapKit

Maps
CoreLocation & MapKit

CoreLocation (1/7)

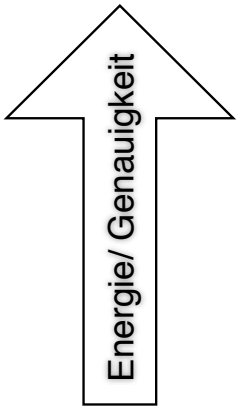
- Wie genau ist die Messung?

```
var horizontalAccuracy: CLLocationAccuracy { get }; // in Meter
var verticalAccuracy: CLLocationAccuracy { get }// in Meter
```

  - Negative Werte zeigen das ein Fehler vorliegt

- Wie kann ich die Messung genauer beeinflussen?

```
kCLLocationAccuracyBestForNavigation //Navigationsapps
kCLLocationAccuracyBest  //default Value
kCLLocationAccuracyNearestTenMeters
kCLLocationAccuracyHundredMeters
kCLLocationAccuracyKilometer
kCLLocationAccuracyThreeKilometers
```



CoreLocation (2/7)

- speed
  ```
  var speed: CLLocationSpeed { get }
  ```
- ```
  //in meters/second !Ungenau
  ```
- course
  ```
  var course: CLLocationDirection { get }
  ```
- ```
  //in Grad 0° Norden/ 90°Osten/ 180° Süden/ 240° Westen
  ```

CoreLocation (3/7)

- timestamp
- @NSCopying var timestamp: NSDate! { get }
  Überprüfen auf zeitliche AbweichungdistanceFromLocation:
- func distanceFromLocation(_ location: CLLocation!) -> CLLocationDistance
  Distanz zwischen den aktuellen Punkt und einer neuen Location in Meter

CoreLocation (4/7)

- CLFloor
- @NSCopying var floor: CLFloor! { get }
  effekitv noch nicht nutzbar/ Indoor Navigation Coming Soon
- CLVisit
  Neue Möglichkeit des location monitoring
- Ein Objekt von CLVisit beinhaltet die Zeit die ein User an einem Ort verbracht hat (coordinate/Start Timestamp / End Timestamp)
  .startMonitoringVisits() //starten der aufzeichnungen
  .stopMonitoringVisits()

CoreLocation (5/7)

- CLGeocoder
  umwandlung einer Location (corrdinaten) in ein Objekt das informationen zu dieser enthält

- func reverseGeocodeLocation(_ location: CLLocation!, completionHandler completionHandler: CLGeocodeCompletionHandler!)

CoreLocation (6/7)

- Permissions
- "When In Use"
- App erhält nur Location Informationen wenn Sie aktiv ist
- "Always"
- App erhält auch im hintergrund Location Informationen

CoreLocation (7/7)

- Code Demo
- Location
- GeodataReverse
- CLVisit
- Notification

MapKit (1/6)

- MKMapView
  - displays a Map
- Map can have annotations on it // Small Views
  MKAnnotation Protocol:
  - var coordinate: CLLocationCoordinate2D { get }
  - optional func setCoordinate(_ newCoordinate: CLLocationCoordinate2D)
  - optional var title: String! { get }
  - optional var subtitle: String! { get }
- Optional Callout for further actions

MapKit (2/6)

- Create an IBOutlet with drag and drop from the Interface Builder

- Adds Annotations to MapView:
  var annotations: [AnyObject]! { get } // get complete list (read-only)

- addAnnotations explicitly to annotations
  func addAnnotation(_ annotation: MKAnnotation!)//add specified Object
  func addAnnotations(_ annotations: [AnyObject]!)// add Array

- Remove Annotations explicitly from annotations
  func removeAnnotation(_ annotation: MKAnnotation!)// remove speficied Object
  func removeAnnotations(_ annotations: [AnyObject]!)// Removes an array of objects from the map view.

MapKit(3/6)

- On Initialising the map it's recommend by apple to add all of your annotation objects to the MapView. Each Objects determines by it selfs when its corresponding annotation view needs to appear onscreen

- The MKAnnotationView handles the layout of a annotation objects on the Map View. By default is the MKPinAnnotationView

Source: MKMapView Class Reference

MapKit (4/6)

- If you touch on an annotation
  - var canShowCallout: Bool // if true
    standard Annotation Shows: title and Subtitle

- If you touch in the bubble the left side / right side
  - var leftCalloutAccessoryView: UIView! // default is nil
    standard: display further information
  - var rightCalloutAccessoryView: UIView! //default is nil
    standard: more detailed information

- Is called when touched:
  - optional func mapView(_ mapView: MKMapView, didSelectAnnotationView view: MKAnnotationView!) // Use this Method to load Information about a Annotation or Images

MapKit (5/6)

- var mapType: MKMapType
  Changes the Layout for the map

- User current Location
  var showsUserLocation: Bool
  should Map Show Users Location
  var userLocationVisible: Bool { get }
  is the User current Location is visible in the map View
  var userLocation: MKUserLocation! { get }
  Annotation Object representing the user

MapKit (6/6)

- var userTrackingMode: MKUserTrackingMode
  MKUserTrackingModeFollow // Map follows User Location
  MKUserTrackingModeFollowWithHeading // Map follows User Location and rotates if heading is changing
  func setUserTrackingMode(_ *mode*: MKUserTrackingMode, animated *animated*: Bool)
  Sets the mode used to track the user location with optional animation.

MKAnnotationView

MKAnnotationView